Modeling Controls for Dynamic Value Exchanges in Virtual Organizations

Yao-Hua Tan[1], Walter Thoen[1], and Jaap Gordijn[2]

[1] Free University, Dept. of Economics and Business Administration, The Netherlands. ytan@feweb.vu.nl
[2] Free University, Dept. of Computer Science, The Netherlands. gordijn@cs.vu.nl

Abstract. The *e*^{3}*-value* modeling tool was developed for the design of a value proposition for virtual organizations. However, it is less suitable for designing the control structure of the virtual organization. We show how *e*^{3}*-value* can be extended using legal concepts such as ownership, possession, usufruct and license. We also introduce value object transfer diagrams that show the transfers of value objects graphically and that can be used for elicitation of the required control mechanisms in order for the virtual organization to function properly and with a level of risk that is acceptable to all parties in the virtual organization.

1 Introduction

Virtual organizations are an important new governance structure for many transactions. A virtual organization can be defined as 'an organization network, which is structured and managed in such a way that it operates vis à vis customers and other external stakeholders as an identifiable and complete organization' [11].

The design of a virtual organization is far from trivial. It is a balancing act between potentially conflicting interests and concerns of participating enterprises. Many approaches suggest that a suitable starting point for designing a virtual organization is the value proposition(s) of such an organization (see e.g. [17] and [3]). Such a proposition contains at least two important elements: (1) a description of the participating actors, the value adding activities they perform and the objects of economic value they exchange with each other and with their customers, and (2) a description of contracts including inter-organizational controls. Since participants in a virtual enterprise do not trust each other on forehand with respect to the objects of economic value they exchange, contracts and supporting controls are used to enable secured participation in a virtual enterprise.

The proposition of a virtual enterprise can be described in many ways, e.g. by natural language as it is often done in practice. In this paper however, we propose a more formal, conceptual modeling, based way to lay down this proposition. Mylopoulos (1992) defines conceptual modeling as 'the activity of formally defining aspects of the physical and social world around us for the purpose of understanding and communication'. Natural language has a few serious drawbacks,

C.D. Jensen et al. (Eds.): iTrust 2004, LNCS 2995, pp. 236–250, 2004.

compared to formal modeling, such as noise (irrelevant information), silence (omission of important information), over specification, contradictions, ambiguity, forward references, and wishful thinking (Mayer 1985). Consequently, we advocate to conceptualize the value proposition thoroughly to create a shared understanding of the proposition at stake. Specifically in the case of virtual enterprises, with involvement of different types of stakeholders representing different interests and concerns of companies the risk of mis-understanding is high. Additionally, (semi-) formal conceptual models allow for proper analysis and provide a starting point for the design of inter-organizational information systems that support the virtual enterprise.

In this paper we present an approach for designing a virtual organization both from a value proposition and trust/control perspective. We first discuss design and life-cycle models of virtual organizations (Sec. 2). These models show that contractual elements in terms of value objects to be exchanged and inter-organization controls are an important tool for structuring virtual organizations. Then we introduce in Sec. 3 the e^3-*value* methodology for conceptualizing a virtual enterprise's value proposition [6,7]. As we will see in Sec. 4, the e^3-*value* methodology is suitable for representing a value proposition but lacks functions for representing trust and associated control issues. To this end, we introduce a new description technique, called a value object transfer diagram, to analyze the vulnerabilities of the members (See Sec. 5). Finally, in Sec. 6, we present our conclusions.

2 Modeling Tools for Designing Virtual Organizations

Let us first look a design approach for virtual organizations in general, before we look at the modeling tools in detail. In [3] Carson et al. present a framework for designing institutions. In this framework the virtual organization is called an Institutional Arrangement (IA) and is distinguished from the Institutional Environment (IE). Carson et al. describe their framework as follows:

'The framework begins with a consideration of the desired outputs and the activity sets required to bring about these outcomes. Then we design contractual, ownership, and social elements of IAs that support these joint profit-maximizing activity sets according to our remediable efficiency tests. We proceed in a staged manner, moving from contractual to ownership to social (relational and reputational) elements of the IA.'

The Carson *et al.* framework is shown in Figure 2. In literature on virtual organizations the assumption is made that a virtual organization rapidly restructures itself if circumstances change. However, it is not obvious when the virtual organization should restructure or when it should remain as is. For example, if restructuring the virtual organization would cause an existing member to be much worse of, then this member would obviously object to the restructuring. Carson *et al.* use 'the remediable efficiency criterion' to evaluate a possible restructuring. The remediable efficiency criterion as defined as:

An IA (and the activity set that it allows) is remediably efficient if it maximizes the joint profit created in an Marketing Value System (MVS) subject to the IA's feasibility given (1) the IE and characteristics of the proposed activity set and (2) switchover costs associated with transitioning into and out of the IA. ([3], page 118)

Hence, only in case joint profit-enhancing actions require reallocation to align efficiency with own-firm profits a new IA comes into consideration. The virtual organization will have to be restructured every time the members have identified this kind of joint profit-enhancing actions. If we assume that this happens frequently, then it is important the elements that make up the structure of the virtual organization are flexible enough.

The assumption that Carson et al. make is that contractual arrangements are the least complex and that social norms are the most complex elements of the structure. Their argument is that 'when feasible, contracting poses the least complex IA design problem because it uses fine-grained support from the IE judiciary, i.e. the system of courts of law, and the IE polity, i.e. the form of political organization, to bind parties to joint profit making (JPM) activity sets" and 'social elements of IAs are ... more complex to develop because they depend minimally on the IE polity and judiciary and almost entirely on norms in the IE and IA to support JPM activity sets'.

Social norms are indeed hard to design as Carson et al. claim. For example, we agree that trust, which is an important concept in the social norms described by Carson et al., is difficult to build or 'design'. However, if (strong) social norms that can support the virtual organization already exist, then it might be easier to rely on these social norms rather than to design new contractual elements. Whether contractual elements can replace social norms is actually a debated topic. For instance, Sitkin and Roth state that 'legalistic remedies have been described as weak, impersonal substitutes for trust' [13]. More dramatically stated, it might be impossible to set up a virtual organizational at all if there is insufficient trust or lack of social norms. Social norms are a much stronger foundation for a virtual organization than contractual norms. In our opinion contractual elements should, therefore, only be used in case no suitable social norms exist. In other words, the contractual elements used to structure the virtual organization should be appropriate to the existing social norms.

3 The *e*^{3}*-value* Methodology

The Carson et al. framework stresses that it should be clear to all members what joint profit the institution is going to achieve and what individual profits each member is going to make. In other words, it is important that the members reach a good and mutual understanding of what the value proposition to the customer is and what value each members contributes to the overall value proposition, what risks and vulnerabilities exist, and how the virtual organization deals with the risks in terms of the control structure of the virtual organization.

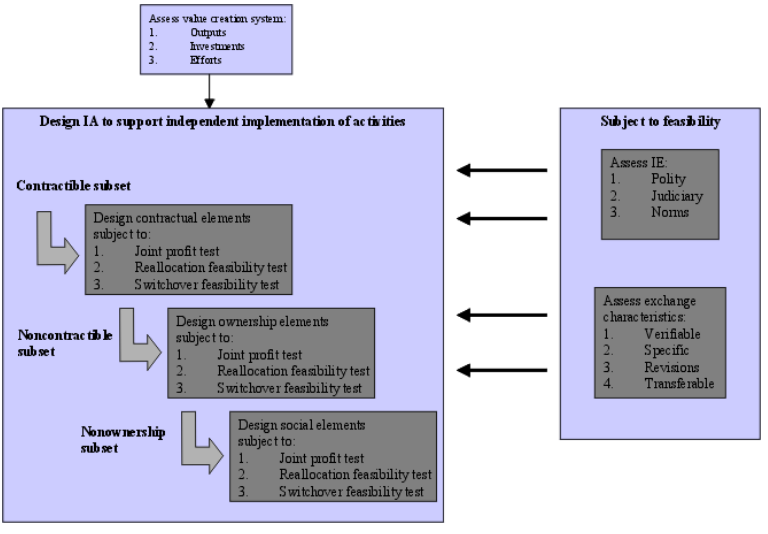

Fig. 1. Design Framework for Institutional Arrangements.

The e^3-*value* theory [6,7] provides a (graphical) conceptual modeling tool for designing and analyzing the value proposition and value exchanges between the members of the virtual organization and between the virtual organization and its customers.

The theory is specifically developed for exploring networked organizations as virtual enterprises are. These organizations tend to be complex in a sense that they consist of many enterprises that offer a joint product or service. Therefore, they are hard to understand at first sight, so it is worthwhile to conceptualize such organizations. Consequently, the e^3-*value* theory has specific modeling constructs to conceptualize *who* offers *what* of economic value to *whom* and expects *what* in return. This contrasts to relatively straightforward examples of e-business such as a single web-shop; the essentials here can be expressed by natural language or forms of structured English that prescribe how to outline a value proposition.

The purpose of the e^3-*value* theory is to provide a *shared* understanding of a virtual enterprise's proposition by thoroughly conceptualizing it. A shared understanding is important because, in practice, the development of a virtual organization's proposition involves a number of persons, all speaking 'different' languages resulting in different interpretations of the proposition. A virtual enterprise consists of other (virtual) enterprises, which in turn are represented by different stakeholders (e.g. CxO's, marketing stakeholders, ICT people and persons dealing with trust issues). The conceptualization constructs (see below) force to ask specific questions to enterprises involved and to steer the discussion to arrive at a shared understanding of the proposition at stake.

Additionally, with tool support it is possible to check whether a proposition is well-formed. An example of such a check is the 'socket'-rule: an enterprise only offers an object of value if and only if s/he obtains another object of economic value in return. Also, it is possible (with tool support) to assess whether a value proposition seems to be profitable for all enterprises involved. In short, we ask enterprises to assign economic value to objects they obtain and deliver, make assumptions on their quantity, and use these numbers to calculate the net cash flow for each enterprise involved (see [7] for more information). At http://www.cs.vu.nl/~gordijn/tools.htm, the reader can download such a tool with these capabilities.

Finally, an e^3-*value* model can be used as a starting point for further analysis and design. In this paper, we use a virtual enterprise's value model for design and assessment of trust- facilitating controls. Other examples are the design of inter-organizational business processes and supporting information systems.

Figure 2 shows an easy to understand e^3-*value* model representing that a supplier offers some object of value to a customer and obtains a fee in return. We keep this value web deliberately simple, to explain our formalization. The grey legend on top of Figure 2 is not part of the e^3-*value* modeling, but is just included to explain the various elements that make up the value model. We now briefly introduce the elements of the e^3-*value* modeling technique (based upon [7]). In the coming sections we discuss the concepts, such as actor, value object and value exchange in more details.

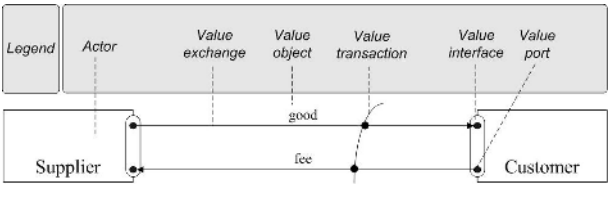

Fig. 2. A supplier and a customer exchanging objects of value. (Note: The grey area and superimposed text are only for explanatory purposes and are not part of the e^3-*value* modeling technique itself)

Actor. An actor is perceived by its environment as an independent economic (and often also legal) entity.

Value Object. Actors exchange value objects, which are services, products, money or even consumer experiences. The important point here is that a value object is of value for one or more actors.

Value Port. An actor uses a value port to show its environment that it wants to provide or request value objects. The concept of port enables us to abstract away form the internal business processes, and to focus only on how external actors can be 'plugged' in.

Value Interface. A value interface models what an actor offers to *and* requests from the environment. Value objects are exchanged via ports, which in

turn are part of a value interface. A value interface assumes atomicity: either *all* ports exchange objects of value or none at all. How this is accomplished is not expressed by e^3*-value* models but a matter of robust process design. This atomicity should in many cases be observed by inter-organizational controls that e.g. can start escalation procedures as soon as atomicity is violated.

Value Exchange. A value exchange is used to connect two value ports with each other. It represents one or more potential trades of value objects between value ports. As such, it is a prototype for actual trades between actors. It shows which actors are willing to exchange value objects with each other.

Value Transaction. A value interface prescribes the value exchanges that should occur, seen from the perspective of an actor the value interface is connected to, because all ports in a value interface should exchange objects, or none at all. Sometimes, it is convenient to have a concept that aggregates all value exchanges, which define the value exchanges that must occur as consequence of how value exchanges are connected, via value interfaces to actors. We call this concept a value transaction. In its simplest form, a transaction is between two actors. However, a transaction can also be between more than two actors. We call such a transaction a *multi-party* transaction.

The e^3*-value* methodology does not tell what specific value exchanges and transactions should be included in the model. Rather, the designer has to go through an elicitation process to find the value exchanges and transactions. It is usually not very hard to list the core value exchanges. For instance, in a sales transaction between a buyer and a seller the core (primary) activities, deliver and pay, will spring to mind immediately and these result in primary value exchanges such as *good* and a *fee*. However, other (secondary) activities such as 'arrange insurance', 'obtain import license' and 'inspect goods' do not immediately spring to mind, but they might be necessary or at least desirable for the successful completion of the primary value transaction.

Secondary activities can be broadly classified in two groups. One group of activities is required to complete the transaction. We call those the 'doing tasks' (see also [1,2]). The other group are activities that are required to monitor the transaction, which we call 'control tasks' (see also [1,2]). The activity 'obtain import license' is an example of a 'doing' task as this activity is required (by some governments) to complete the transaction. The activities 'arrange insurance' and 'inspect goods' are control tasks as these activities are not required to complete the transaction. These activities can be included in the model of the transaction in order to assure that the transaction completes to everyone's satisfaction. In other words, the control tasks are included to alleviate problems that could result from the actions of the members or that could be the result of outside forces (such as the weather, political decisions or technical failures).

The problems that the members foresee with respect to the actions of the members are usually a trust issue. For example, a member is not sure that another member is capable of delivering the required value (e.g. the quality might be an issue) or that the other member might not be willing to delivery the right value under all circumstances (e.g. when there is room for opportunism). Mayer

et al. define trust as 'The willingness of a party to be vulnerable to the actions of another party based on the expectation that the other party will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party' [10]. Hence, in case a member is not willing to be vulnerable to the actions (or a particular action) of another members then we have a trust issue. It is an accepted principle in the literature that a lack of trust can be compensated by the introduction of control mechanisms (see for instance [4,8,13,14,15,16]). In this paper we focus on the design of the control tasks and we use value object transfer diagrams to scan the value exchanges for vulnerabilities.

4 Value Modeling from a Control Perspective

In the previous sections we discussed the importance of contractual control elements for the design of virtual organization. Modeling the trust and control aspects of the contract using (graphical) modeling tools based on the e^3-*value* theory, before or during the drafting of the contract, can improve that quality of contract and can facilitate the negotiation process between the members of the virtual organization. In this section we introduce the general concepts that can be used from a *control* perspective to model business transactions.

We have attempted to stay as close as possible to the definitions in the e^3-*value* ontology [6,7]. However, we have evaluated the e^3-*value* definitions from control modeling perspective. As a result, we introduce several extensions. We will use the following general concepts: actor, role, value object and value object transfer.

Actor. We define the concept of actor in the same way as in the e^3-*value* theory: 'An actor is perceived by his/her environment as an economically independent (and often also legal) entity.' Enterprises and end-consumers are examples of actors. A profit and loss responsible business unit, which can be seen as economically independent is an actor, although such a unit is not a legal entity.

Value Activity and Role. While designing controls, it is convenient to model business transactions without knowing which individual agent is going to participate in that transaction yet. To that end, the e^3-*value* theory proposes *value activity*. This is defined as an activity, which is beneficial for at least one actor. The latter is important, because we want to assign activities to performing actors, and so at least one actor should be interested in the execution of such an activity.

Instead of *value activity* we use the concept of *role* from the research on inter-organizational trust building from Bons[1]: 'A *role* is a model of a meaningful cluster of external activities, recognized by the business world'. With external activities Bons refers to those actions that can be observed by other actors. Typical roles in the business world are buyer, seller, bank, freight forwarder, insurer etc. An actor is said to perform a role, implying that in the business transaction the actor will perform the actions that the role comprises. In our

ontology an actor is always associated with a role. A role, however, can be used without any actor being associated with it.

Value Object. In the *e*^{3}*-value* ontology a value object is defined as follows: 'A value object is a service, a product, or even an experience, which is of economic value for at least one of the actors'. We extend this definition because we want to explicitly include rights, and in particular intellectual property rights, such as copyrights, patents and trademarks, in the definition. Intellectual property rights have always played an important role in many business transactions. Moreover, with the capability to distribute or deliver certain products by means of digital networks we believe that recognizing intellectual property rights and related legal constructs, such as licenses, in value modeling and trust modeling this is becoming even more important. In *e*^{3}*-value* ontology a value object has a *name* as the only property. We add *type* as property in our ontology. The Type property can have the following values: product, service, right and experience. We also extend the ontology by adding as properties some legal constructs that are usually associated with a value object. These legal constructs are: *Ownership*, *Possession*, *Usufruct* and *License*. The following three constructs are used for value objects of type products, and are defined as follows (Webster online):

Ownership: the state, relation, or fact of being an owner (see http://www.webster.com/cgi-bin/dictionary?book=Dictionary).

Possession: a) the act of having or taking into control b) control or occupancy of property without regard to ownership

Usufruct: the legal right of using and enjoying the fruits or profits of something belonging to another

An example of the distinction between these three concepts is that you can be the owner of a house, the tenant can have possession of the house, and a third party can have the usufruct of the house by being entitled to the rent.

These three legal constructs are a property of the value object concept in our model. The value of these properties is always a reference to an actor or a role. In other words, what is important in our ontology is which actor or role has ownership, possession or usufruct of a value object.

For value objects of the type Right we use the constructs Ownership and Usufruct in the same way as for products. In addition we use the construct License in respect to rights. We define a license as follows (based on Webster): "*License*: a permission granted by a competent authority to engage in an activity otherwise unlawful."

Services and experiences cannot be owned or possessed or licensed. A service or an experience can be based on rights owned or licensed. For example, a certain service can be so unique that it can be patented and hence can only be offered by the patent owner. However, we will not say that the actor owns the service in case he owns the patent. The same holds for experiences. Of course, this changes nothing about the fact that a value can be assigned to a service and experience. The value of the service or experience can be exchanged for another value object.

Value Object Transfer. The legal constructs associated with value objects are important for modeling control structures, because business transactions are usually about the exchange of value objects between actors. The legal constructs allow us to be more precise about what we mean with 'the exchange of value objects' as we can now distinguish between a transfer of ownership, a transfer of possession, a transfer of usufruct and a transfer of a license.

An exchange of a value object can of course be any combination of these transfers. The simplest exchange is the exchange in which all transfers occur together. In international trade the transfer of ownership and the transfer of possession, however, are often separated. Typically in the Letter of Credit control procedure, the seller of a value object can first transfer possession to a freight forwarder and retain the ownership and then later, e.g. after receiving payment, transfer the ownership to the buyer. After becoming the owner the buyer can request a transfer of possession from the freight forwarder. The Bill of Lading, which is issued by the forwarder when he receives the goods from the seller, is a control document in the letter of credit procedure that is used to prove that the seller has transferred the possession of the goods to the forwarder.

Many interesting situations exist in which a product is sold in combination with a license. Typical examples are CDs and DVDs. CDs and DVDs are physical products combined with a license to play music or watch a movie. However, it is important to distinguish the license from the product. The same physical DVD is combined with a *license for personal use* when a consumer buys it in a store and is combined with a *rental license* when sold to a video rental shop (usually in return for a much higher fee in the latter case). In Figure 4 the e^3-*value* model of a DV rental scenario is shown.

Note, however, that by renting a DVD from the shop the consumer gets a license for personal use from the copyright owner not from the shop! When an actor without the special rental license lends a DVD to another actor (i.e. only the possession of the physical object is transferred), then no license for personal use is granted by the copyright owner. When the new possessor would watch the DVD, then he would violate the copyright! In the next session we introduce value object transfer diagrams that allow us to model these details in a more precise manner.

5 Value Object Transfer Diagram

We use a Value Object Transfer Diagram to visualize the value object transfers. The diagram consists of roles, transfers and tokens. The colored tokens represent the concepts ownership, possession, usufruct, license and value. The transfers represent the transfer of one or more concepts.

An important distinction is that e^3-*value* models are static models whereas the value object transfer models are dynamic. The e^3-*value* model shows the value exchanges between the roles without modeling the actual flow of value through the model. The value object transfer diagram uses the concept State

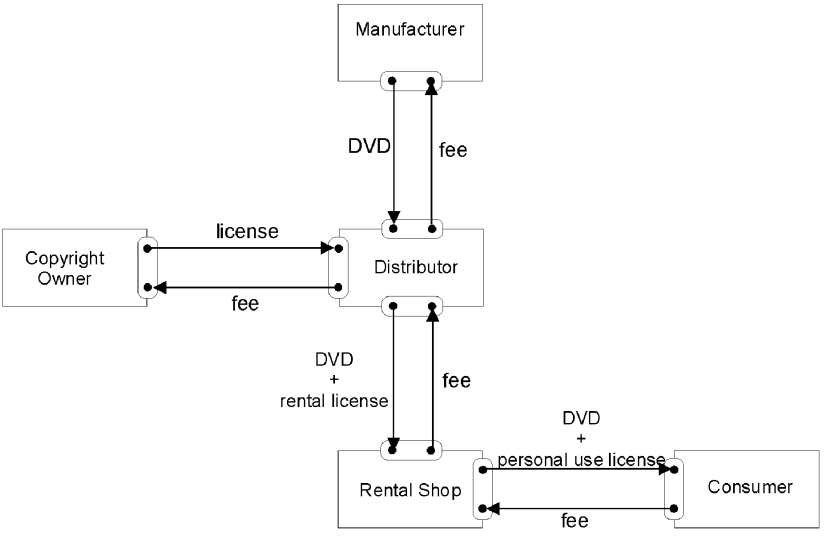

Fig. 3. An e^3-*value* model of the DVD rental scenario

for capturing the dynamics. A state is defined as any distribution of the tokens over the roles.

The idea is that the diagrams are used to graphically show the transfers of value objects that take place in a business transaction whilst leaving the sequencing open for further planning. Note that the difference between for instance a 'pre-payment' scenario and a 'post-payment' scenario is the sequencing of the transfers, and not the transfers that make up the business transaction. The sequencing is left for a later stage because the scenarios are quite different from trust perspective. For example, in a typical post-payment scenario goods are delivered before any money is received in return. If there are no controls in place then the party delivering the goods needs to have sufficient trust in the other party paying eventually in order for the parties to be able to agree on this scenario. Whether the trust levels are sufficient or not, is a question that needs to be answered at a later stage of our approach. At that stage there might be a negotiation between the parties and controls might have to be added in order to reach a satisfactory agreement. The intention of the diagrams presented here is to make the parties aware of the trust issue related to this particular value exchange.

Note that this is also an important distinction from process modeling tools, such as Petri Nets, state-transition diagrams or workflow management systems. In the process modeling tools the sequencing of actions is very important. This makes such models very useful for specifying the execution of transactions, but not for the trust design that is the subject of this paper.

Figure 5 shows a Value Object Transfer Diagram for the simple international trade example described above. Initially, all tokens, i.e. value object concepts, are with role 1, i.e. the seller. Then transaction t1 fires and the possession is transferred to role 2, i.e. the freight forwarder. Then transaction t2 fires and the ownership and usufruct are transferred to role 3, i.e. the buyer. And finally transaction t3 fires and the possession is transferred to role 3, i.e. the buyer. Note that in international trade the transition t2 usually takes the form of sending the Bill of Lading to the buyer and that transition t3 requires the buyer to show the bill of lading to the freight forwarder.

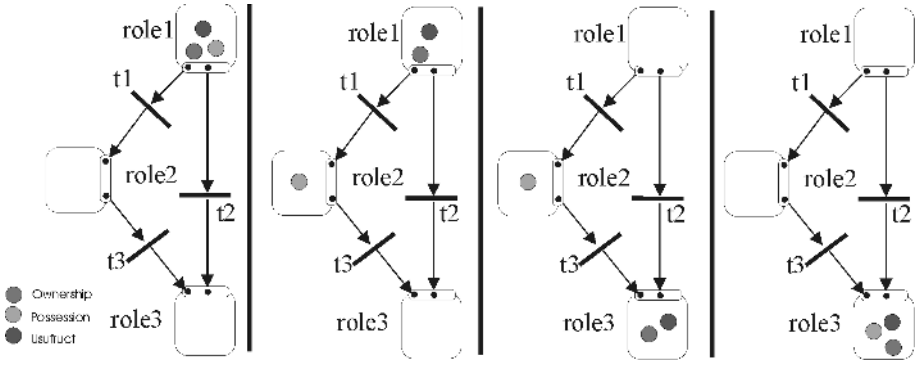

Fig. 4. Value Object Transfer Diagram.

The value object transfer diagram as shown in Figure 5 is a mix of elements from the e^3-*value* models and of Colored Petri Nets. To make this mix possible of these two representation tools we will not assume the Petri Net property that it must an a-cyclic graph, i.e. that there should be no loops. We will later explain why we need these cycles, and also how it can be easily repaired in order to get a proper Petri Net. The most important reasons that we allow for cycles is that a value object can be transferred back and forth between two or more roles. The value object transfer diagrams can be converted into The value interface and value ports are taken from the e^3-*value* models and the colored tokens and transitions and arcs are taken from colored petri nets.

Also note that Figure 5 is obviously not a complete transaction. For example, the payment, i.e. the transfer of ownership of a monetary value object, from the buyer to the seller is not included in this figure. The freight forwarder has to be paid by either the seller or the buyer for his service too. The service provided by the freight forwarder is also a value object according to our definition. As it is unintuitive to talk about the ownership or possession of a service or an experience, we will simply refer to a 'transfer of value' in relation to value objects of the type service and experience.

The issue of ownership is not fully covered by existing international conventions and can thus differ in the various national laws [18]. For instance for the transfer of ownership many national laws make a distinction between register goods, such as houses and cars and non-register goods. The legal requirements related to these distinctions might differ among the national laws though. In the Value Object Transfer diagram we will not take any specific legal requirements into account regarding a transfer of ownership, we simply assume that a transfer of ownership occurs or does not occur and if it occurs it is always successful.

Note again that the Value Object Transfer Diagram is quite different from the e^3-*value* models. The purpose of the Value Object Transfer Diagram is to model how the value objects flows from one actor/role to another actor/role during the execution of a transaction. The purpose of the e^3-*value* models is to model what value objects actors offer in exchange for other value objects. Moreover, the e^3-*value* models assume that the exchange of value objects is atomic at the level of the value interface. According to Gordijn, 'This ensures that if an actor offers something of value to someone else, s/he always gets in return what s/he wants. How this is ensured is a matter of a robust business process design, legal agreements, or sometimes use of technology, but this is not of interest for the value model'([5], page 53). Designing control mechanisms to ensure atomicity is exactly what the research described in this paper aims to accomplish. Hence, in order to support the design of robust business processes with the appropriate control structure, after the business opportunity has been clearly established using e^3-*value* models in the first phase of the life-cycle, we have to assume that the exchange of value objects is not atomic at the level of the value interface.

In Figure 5 a part of the DVD rental example is shown in a value object transfer diagram. The figure represents the part of the scenario in which the consumer rents a DVD from the rental shop and later returns the DVD, i.e. the value transfers for the manufacturer and distributor are not shown.

The figure shows three states 1) the begin state before the DVD is rented, 2) the intermediate state in which the consumer has rented the DVD and 3) the final state in which the consumer has returned the DVD.

There are six value object transfers. In the first transfer t1, the rental shop transfers the possession of the DVD to the consumer in return for which the consumer transfers the ownership, possession and usufruct of some money to the rental shop (transfer t2). At the same time the copyright owner transfers a person use license to the consumer (transfer t3). We are now in the intermediate state where the consumer has the possession of the DVD and a personal use license to watch it. Note that the copyright owner has not received anything in return for transferring the license. This is because the copyright owner has received a fee beforehand from the distributor, which is not shown in the diagram.

When the consumer returns the DVD, i.e. the consumer transfers the possession of the DVD back to the rental shop (transfer t5) and at the same time the consumer returns the license to the copyright owner (transfer t4). Note that we would have a trust problem if the consumer would get nothing in return for transfer t4 and transfer t5 and the scenario would terminate after transfer 5. If

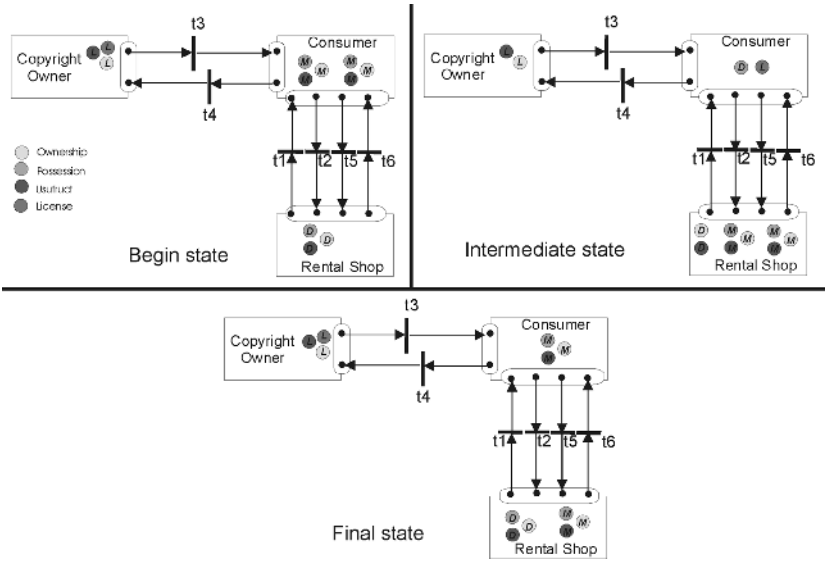

Fig. 5. Value Object Transfer diagram for DVD example.

there was no incentive for the consumer to execute transfers t4 and t5, then why would he do so? This is where the control structure becomes important.

There are several possible control mechanisms that we could introduce to provide an incentive for the consumer to return the DVD. First of all, the rental shop, copyright owner and consumer should have a contract governing the entire exchange. The contract should state that the consumer is obliged to return the possession of the DVD and the license at some point. The institutional environment is of course important as the IE might have to enforce this contract. However, the strength of the incentive for the consumer to return the DVD depends on how the IE would enforce the contract. If the IE would just make the consumer return the DVD without a penalty, then the consumer could easily decide to wait for this to happen. Therefore, an additional control mechanism to ensure a prompt return of the DVD would be a penalty clause in the contract, which stipulates a (monetary) penalty for a late return.

In Figure 5, however, we have opted for another control mechanism; namely a deposit. We assume that in transfer t2 the amount of money paid by the consumer covers two things: 1) the fee for renting the video and 2) a deposit (note that in Figure 5 there are two monetary value objects). Therefore, there is a final transfer t6 in which the rental shop transfers the deposit back to the consumer after the transfers t4 and t5 have occurred.

The value object transfer diagrams can support the designer to find the transfers that might require control mechanisms and to model the relevant value transfer aspects for the design of appropriate control mechanisms. The DVD

rental example, which at first appears to be very simple, turns out to be quite complex if we model the actual value object transfers and control mechanisms.

As we mentioned above the value object transfer diagrams are not always a-cyclic. In particular, in the above figure many of the arcs constitute loops between actors. This is not in accordance with the basic definition of Petri nets that it should be a-cyclic graphs. However, the value object transfer diagrams can be easily converted into colored Petri nets. The role places will have to be separated in several places in the colored petri net, ensuring that a token can never be in the same place more than once. For example, transfer t3 and t4 constitute a cycle. But if we would represent the actor Copyright Owner twice, then we could remove this cycle by first having transfer t3 from Copy Right to Consumer and then subsequently transfer t4 from Consumer to the second instance of Copy Right Owner. A similar approach also works for the other cycles. This also shows that an explicit representation of the actual process flow would become much more complicated, and less insightful for modeling the value exchanges and corresponding control mechanisms. The morale here is, as with most representation formalisms, that the correctness of a representation formalism critically depends on the modeling objective. Modeling value exchanges apparently requires a slightly different perspective than pure process modeling.

Here we extended e^3-*value* with a kind of Petri nets to represent the dynamic transfer of the legal constructs. Other formalisms are also widely used to model legal aspects. For example, there is a long tradition of deontic logic to model legal notions (see e.g. [9]). However, most of these formalisms focus mainly on the representation of static legal aspects, whereas we focused on the dynamic aspects of the transfer of legal constructs. In [12] a deontic logic is discussed based on dynamic logic. In future research we plan to investigate the relation between our value object transfer diagrams and this deontic this logic.

6 Conclusions

In this paper we have discussed the importance of designing the control structures of virtual organizations and the importance of having the right modeling tools for this task. The e^3-*value* theory provides a (graphical) modeling tool for elicitation of the value proposition of the virtual organization and the value exchange between the members of the virtual organization. We have shown that the e^3-*value* theory, however, does not allow us to design the control structure of the virtual organization which in our opinion can be equally important to a proper value proposition. The e^3-*value* theory is intended to model a virtual enterprise's proposition from a business value perspective only. Consequently the theory is not detailed enough about the actual transfer of value objects. To overcome these constraints on the e^3-*value* theory we introduced models that enable a more detailed analysis of the value object transfers. We used important legal constructs such as ownership, possession, usufruct and license to achieve this. We also introduced the idea of value object transfer diagrams to model the transfers graphically.

References

1. R.W.H. Bons. *Designing Trustworthy Trade Procedures for Open Electronic Commerce*. PhD thesis, EURIDIS and Department of Business Administration, Erasmus University, Rotterdam, NL, 1997.
2. R.W.H. Bons, R.M. Lee, and R.W. Wagenaar. Designing trustworthy interorganizational trade procedures for open electronic commerce. In *Global Business in Practice, Tenth International Bled Electronic Commerce Conference, Bled, Slovenia*, 1997.
3. S.J. Carson, T.M. Devinney, G.R. Dowling, and G. John. Understanding institutional designs within marketing value systems. *Journal of Marketing*, 63:115–130, 1999.
4. T.K. Das and B.S.Teng. Between trust and control: developing confidence in partner cooperation in alliances. *Journal of Decision Support Systems*, 23(3):491–512, 1998.
5. J. Gordijn. *Value-based Requirements Engineering - Exploring Innovative e-Commerce Ideas*. PhD thesis, Vrije Universiteit, Amsterdam, NL, 2002. Also available from http://www.cs.vu.nl/~gordijn/.
6. J. Gordijn and J. M. Akkermans. Designing and evaluating e-Business models. *IEEE Intelligent Systems - Intelligent e-Business*, 16(4):11–17, 2001.
7. J. Gordijn and J.M. Akkermans. Value-based requirements engineering: Exploring innovative e-commerce ideas. *Requirements Engineering Journal*, 8(2):114–134, 2003.
8. S.D. Jap and S. Ganesan. Control mechanisms and the relationship life-cycle. *Journal of Marketing Research*, 1998.
9. A. Jones and M. Sergot. Deontic logic in the representation of law: Towards a methodology. *Artificial Intelligence and Law*, 1(1), 1992.
10. R.C. Mayer, J.H. Davis, and F.D. Schoorman. An integrative model of organizational trust. *Academy of Management Review*, 20(3):709–734, 1995.
11. M. Mazzeschi. What is a virtual enterprise? In *presentation ESoCE*, 2000.
12. J-J.Ch. Meyer, R.J. Wieringa, and F.P.M. Dignum. The role of deontic logic in the specification of information systems. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*, pages 74–81, Boston/Dordrecht, 1998. Kluwer.
13. S.B. Sitkin and N.L. Roth. Explaining the limited effectiveness of legalistic 'remedies' for trust/distrust. *Organization Science*, (4):367–392, 1993.
14. Y.H. Tan and W. Thoen. Toward a generic model of trust for electronic commerce. *International Journal of Electronic Commerce*, 5(2):61–74, 2000.
15. Y.H. Tan and W. Thoen. Formal aspects of a generic model of trust for electronic commerce. *Journal of Decision Support Systems*, 33:233–246, 2002.
16. Y.H. Tan and W. Thoen. Electronic contract drafting based on risk and trust assessment. *International Journal of Electronic Commerce*, to appear, 2003.
17. Virtual Vertical Enterprise (VIVE). *VIVE Final Report*. ESPRIT 26854.
18. A. von Ziegler, J.H. Ronoe, C. Debattista, and O.B. Plegat-Kerrault. *Transfer of Ownership in International Trade*. CC publication 546, ICC publishers, Paris.